Name: Satyam Rai
UID: 2019130051
TE COMPS
Batch C

EXPERIMENT 4

(Naive Bayes)

Aim: To train a machine learning model using naïve bayes algorithm to classify whether a given mail is spam or not.

Code:

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB

df=pd.read_csv('spam1.csv')

df.head()

df.shape

df.sort_index(inplace=True)

df.tail()

v = CountVectorizer()

all_features = v.fit_transform(df.Message)

all_features.shape

v.vocabulary_
```

```
x_train, x_test, y_train, y_test =
train_test_split(all_features,df.Category,test_size=0.3,random_state=88)

x_train.shape

classifier = MultinomialNB()
classifier.fit(x_train,y_train)
print(f'Accuracy is: {classifier.score(x_test,y_test):.2%}')

if classifier.predict(v.transform(['WINNER!! you have a chance to win a free Iphone
now']))==[1]:
    print('SPAM')
else:
    print('NOT SPAM')
```

Output:

```
if classifier.predict(v.transform(['WINNER!! you have a chance to win a free Iphone now']))==[1]:
    print('SPAM')
else:
    print('NOT SPAM')
```

```
SPAM
```

Conclusion:

In this experiment, the aim was to implement the Naive Bayes Algorithm to train a machine learning model which helps to predict whether a given mail is spam or not. The dataset used for the preparation of the model is available above. The dataset contains numerous mail bodies. To train the model, first the body of the mail was split into individual words and these words are then considered as discrete features which are then used to train the model and the data set was split into an 70:30 ratio and was trained on the 70% of data and the rest was kept as test data.
Using the Naive Bayes Algorithm, the data set was trained and the corresponding labels were predicted. Then using the test data, I calculated the accuracy of the model trained which came around 97%.